

Εργασία 3 – Δικτυακό σύστημα αρχείων

Υλοποιήστε ένα απλό δικτυακό σύστημα αρχείων που επιτρέπει σε διεργασίες να δημιουργούν και να προσπελάζουν με διαφανή τρόπο απομακρυσμένα αρχεία σε ένα άλλο μηχάνημα εξυπηρετητή. Η επιθυμητή διαφάνεια πρέπει να επιτυγχάνεται μέσω ενδιάμεσου λογισμικού που παρέχει στην εφαρμογή κατάλληλες λειτουργίες στο πνεύμα του εξής API:

int mynfs_set_srv_addr(char *ipaddr, int port);	Στοιχεία επικοινωνίας εξυπηρετητή.
int mynfs_set_cache(int size, int validity);	Προσδιορισμός μεγέθους κρυφής μνήμης και διάρκειας εγκυρότητας δεδομένων.
int mynfs_open(char *fname, int flags);	Άνοιγμα/δημιουργία αρχείου για διάβασμα/γράψιμο, ανάλογα με τα flags (O_CREAT, O_EXCL, O_TRUNC, O_RDWR, O_RDONLY, O_WRONLY).
int mynfs_read(int fd, void *buf, size_t n);	Διάβασμα δεδομένων από το αρχείο.
int mynfs_write(int fd, void *buf, size_t n);	Γράψιμο δεδομένων στο αρχείο.
int mynfs_seek(int fd, off_t pos, int whence);	Μετακίνηση θέσης ανάγνωσης/εγγραφής, ανάλογα με την τιμή του whence (SEEK_SET, SEEK_CUR, SEEK_END).
int mynfs_close(int fd);	Κλείσιμο αρχείου.

Η υλοποίηση αποτελείται από δύο τμήματα: το λογισμικό του πλετάτη και το λογισμικό του εξυπηρετητή. Η αλληλεπίδραση ανάμεσα τους πρέπει να γίνει με βάση τα εξής RPCs: (1) int openRPC(char *fname, int flags, int *fid) για τον εντοπισμό/δημιουργία ενός αρχείου και την επιστροφή του εσωτερικού αναγνωριστικού του, (2) int readRPC(int fid, int pos, void *buf, int *nbytes) για την ανάγνωση δεδομένων από το αρχείο, και (3) int writeRPC(int fid, int pos, const void *buf, int nbytes) για την εγγραφή δεδομένων στο αρχείο. Τα RPCs πρέπει να υλοποιηθούν πάνω από UDP/IP με σημασιολογία at-least-once, και έτσι ώστε να μην χρειάζεται ο εξυπηρετητής να κρατά ξεχωριστή κατάσταση για κάθε πελάτη.

Κάθε φορά που η εφαρμογή ανοίγει ένα αρχείο μέσω mynfs_open(), το λογισμικό πελάτη επιστρέφει έναν νέο file descriptor (όπως κάνει και το λειτουργικό σύστημα) που στην συνέχεια μπορεί να χρησιμοποιηθεί ως παράμετρος για τις υπόλοιπες λειτουργίες πάνω στο αρχείο. Το λογισμικό πελάτη πρέπει να κάνει τους απαραίτητους ελέγχους με βάση και τα flags που χρησιμοποιήθηκαν στην mynfs_open(), και να εκτελεί τα κατάλληλα RPCs προς τον εξυπηρετητή, με διαφανή τρόπο για την εφαρμογή. Ο εξυπηρετητής διαχειρίζεται έναν (για απλότητα) επίπεδο χώρο ονομάτων στο τοπικό σύστημα αρχείων, με ρίζα έναν κατάλογο που δίνεται ως όρισμα κατά την εκκίνηση του. Εσωτερικά, αντιστοιχίζει κάθε αρχείο που προσπελάζεται σε ένα αριθμητικό αναγνωριστικό (file id) που χρησιμοποιείται ως αναφορά στο πλαίσιο των υπόλοιπων αλληλεπιδράσεων (RPCs) με τους πελάτες, έτσι ώστε να μην ξαναστέλνεται το όνομα του αρχείου πάνω από το δίκτυο. Σκεφτείτε τι πρόβλημα δημιουργεί η χρήση τέτοιων αναγνωριστικών, και πως αυτό μπορεί να λυθεί με πρακτικό τρόπο.

Για να επιταχυνθεί η προσπέλαση στα απομακρυσμένα αρχεία, το λογισμικό του πελάτη κρατά σε κρυφή μνήμη τμήματα του αρχείου ώστε αυτά να είναι άμεσα διαθέσιμα αν τα ξαναζητήσει η εφαρμογή. Ο έλεγχος εγκυρότητας των δεδομένων της κρυφής μνήμης πρέπει να υλοποιηθεί στο πνεύμα του NFS, με το μέγεθος και το χρονικό διάστημα εγκυρότητας των δεδομένων να καθορίζεται από την εφαρμογή μέσω της mynfs_set_cache(). Για ευκολία, το λογισμικό του πελάτη μπορεί να λαμβάνει τα περιεχόμενα των αρχείων μέσω του readRPC() σε τμήματα σταθερού μεγέθους, ανεξάρτητα από το πόσα bytes ζητά να διαβάσει η εφαρμογή. Όταν εξαντληθεί ο χώρος της κρυφής μνήμης, πρέπει να εκδιώκονται ένα ή περισσότερα τμήματα με βάση μια πολιτική που θα αποφασίσετε εσείς. Στο πνεύμα του write-through, ο πελάτης μεταδίδει άμεσα στον εξυπηρετητή τις αλλαγές που κάνει η εφαρμογή στο αρχείο, και ο εξυπηρετητής περνά τις αλλαγές άμεσα στον δίσκο.

Κάντε κατάλληλες δοκιμές για να επιβεβαιώσετε την ορθή λειτουργία της υλοποίησης σας, π.χ., μέσα από μια εφαρμογή που δημιουργεί ένα τοπικό και ένα απομακρυσμένο αντίγραφο ενός απομακρυσμένου αρχείου και στην συνέχεια συγκρίνει τα περιεχόμενα τους. Επίσης, αξιολογήστε την βελτίωση της απόδοσης που επιφέρει η χρήση της κρυφής μνήμης για διαφορετικά σενάρια ανάγνωσης/γραφίματος ενός αρχείου και μεγέθους κρυφής μνήμης.

Χρησιμοποιήστε όποια γλώσσα προγραμματισμού επιθυμείτε. Ο κώδικας σας πρέπει να μεταφράζεται και να εκτελείται κανονικά στο περιβάλλον Linux του εργαστηρίου. Εναλλακτικά, μπορείτε να επιδείξετε την υλοποίηση σε δικά σας laptop που θα πρέπει να συνδέονται στο ενσύρματο δίκτυο του εργαστηρίου. Σιγουρευτείτε ότι η επίδειξη σας δουλεύει σωστά. **Δεν μπορείτε να κάνετε καμία υπόθεση για τα περιεχόμενα των αρχείων που διαβάζει/γράφει η εφαρμογή.**

Φροντιστήριο/συζήτηση: Πέμπτη 4 Απριλίου 2019 Ημερομηνία παράδοσης: Σάββατο 20 Απριλίου 2019, 22:00
